THE COMMISSIONER IS AUTHORIZED TO CHARGE ANY DEFICIENCY IN THE FEES FOR THIS PAPER TO DEPOSIT ACCOUNT NO. 23-0975

#3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of :

Toshihiro WAKAYAMA :

Serial No. NEW : **Attn: APPLICATION BRANCH**

Filed March 20, 2001 : Attorney Docket No. 2001_0284A

A SYSTEM FOR MANAGING NETWORKED INFORMATION CONTENTS

CLAIM OF PRIORITY UNDER 35 USC 119

Assistant Commissioner for Patents,
Washington, DC 20231

Sir:

Applicant in the above-entitled application hereby claims the date of priority under the International Convention of Japanese Patent Application No. 2000-091455, filed March 29, 2000, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Toshihiro WAKAYAMA

By _____
Charles R. Watts
Registration No. 33,142
Attorney for Applicant

CRW/asd
Washington, D.C. 20006
Telephone (202) 721-8200
Facsimile (202) 721-8250
March 20, 2001

日　本　国　特　許　庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年　３月２９日

出願番号
Application Number: 特願２０００－０９１４５５

出願人
Applicant (s): 若山　俊弘

２００１年　１月２６日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２００１－３００１９４６

【書類名】　特許願
【整理番号】　P561
【あて先】　特許庁長官殿
【発明者】
　【住所又は居所】　新潟県南魚沼郡大和町大字市野江甲３５
　【氏名】　若山　俊弘
【特許出願人】
　【住所又は居所】　新潟県南魚沼郡大和町大字市野江甲３５
　【氏名又は名称】　若山　俊弘
【代理人】
　【識別番号】　100071320
　【弁理士】
　【氏名又は名称】　田辺　敏郎
【手数料の表示】
　【予納台帳番号】　014317
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　情報コンテンツの連携管理システム

【特許請求の範囲】

【請求項１】　ウェブ上で相互に連携し依存する情報コンテンツの集合を管理する連携管理システムにおいて、該システムはコンピュータによる解釈実行が可能な形式で記述された要素間の依存関係を伴うウェブ・ドキュメントの集合であるコンテンツ・ネットを備え、一部の要素にコンテンツ変更が生じた場合に変更された要素に依存するすべての要素をコンピュータによって自動的に連鎖更新することを特徴とする情報コンテンツの連携管理システム。

【請求項２】　コンテンツ・ネットにおける要素間の依存関係が記述された制約ファイルと前記依存関係の対象となるウェブ・ドキュメントであるコンテンツ・ファイルとが別ファイルとして存在することを特徴とする請求項１記載の情報コンテンツの連携管理システム。

【請求項３】　コンテンツ・ネットにおけるコンテンツ・ファイルと、そのコンテンツ・ファイル内の要素が依存する他の要素との依存関係が記述された制約ファイルとが対になって存在することを特徴とする請求項２記載の情報コンテンツの連携管理システム。

【請求項４】　コンテンツ・ネット内の全てのコンテンツ・ファイルにおける依存関係の対象となる全ての要素の集合に対し、その集合内の各要素がその集合において独自の名称であるコンテンツ変数、あるいはそれと同等の識別形式である識別子を与えられ、前記要素間の依存関係が前記コンテンツ変数を含む前記識別子の間の関係としてＸＭＬ（Extensible Markup Language）及びＭａｔｈＭＬ（Mathematical Markup Language）などのウェブ標準言語で記述されることを特徴とする請求項１から請求項３のいずれかに記載の情報コンテンツの連携管理システム。

【請求項５】　依存関係がコンテンツ変数間の関係として記述され、該コンテンツ変数間の関係が、コンテンツ変数を表す第１項、コンテンツ変数の集合を表す第２項及び第１項と第２項の関係を表す依存定義の第３項からなる依存節、若しくはそれと同等な内容をもつ他の形態として制約ファイルに記述されるか、

又はウェブ・ドキュメントに情報コンテンツとともに混在して記述されることを特徴とする請求項４記載の情報コンテンツの連携管理システム。

【請求項６】　依存節が、第１項のコンテンツ変数と、第２項の各コンテンツ変数に対応する要素間に木構造上の階層関係のない関数依存節と、第１項のコンテンツ変数と第２項の各コンテンツ変数に対応する要素が木構造上の階層関係にある階層依存節からなることを特徴とする請求項５記載の情報コンテンツの連携管理システム。

【請求項７】　関数依存節の第３項が、ＭａｔｈＭＬで提供される演算子、ＭａｔｈＭＬで参照されうる外部演算子及び第２項のコンテンツ変数の関数的構成で表現されることを特徴とする請求項６記載の情報コンテンツの連携管理システム。

【請求項８】　連鎖更新のシステムが、更新のあった要素の集合のいずれかの要素に対応するコンテンツ変数に依存するすべてのコンテンツ変数の集合である更新候補変数集合を決定し、さらに自己依存のサイクルの有無を決定する依存構造分析モジュールを持つ請求項１から請求項７のいずれかに記載の情報コンテンツの連携管理システム。

【請求項９】　更新候補変数集合内のコンテンツ変数を順次、再帰的に更新していくモジュールを持つことを特徴とする請求項８記載の情報コンテンツの連携管理システム。

【請求項１０】　更新候補変数の更新順位を表す更新ランクを決定するモジュールである更新ランクモジュール及び前記更新候補変数を更新ランクに沿って更新するモジュールを持つことを特徴とする請求項８記載の情報コンテンツの連携管理システム。

【請求項１１】　コンテンツ・ネットにおける情報コンテンツを記述する各ウェブ・ドキュメントが、ウェブ・ブラウザで作動する情報送受信のポート複数個の集合からなるポート複合体であるステーションとしてユーザに提示され、前記コンテンツ・ネットが前記ステーションの集合であるステーション・ネットとしてウェブ・ブラウザを通して利用されることを特徴とする請求項９又は請求項１０記載の情報コンテンツの連携管理システム。

【請求項１２】　ステーションが、そのステーションの属するステーション・ネット内からの情報を参照する内部参照ポート、データベースなど外部のアプリケーションからの情報を参照する外部参照ポート、そのステーションの属するステーション・ネット内の他のポートに送信するための送信ポート、そのステーション固有のメモ情報として保有、閲覧される情報を入力するためのローカルポートを備えることを特徴とする請求項１１記載の情報コンテンツの連携管理システム。

【請求項１３】　ステーションにおける各ポートのコントロール機能が、そのステーションを含むステーション・ネット内の連鎖的コンテンツ更新を指示する内部更新機能、データベースなど外部のアプリケーションから情報を取り込むためのインポート機能、外部のアプリケーションに情報を送信するエクスポート機能、あるいはこれら機能の逐次的又は並列的組合せからなるコントロール機能を持つことを特徴とする請求項１２記載の情報コンテンツの連携管理システム。

【請求項１４】　コントロール機能が、人によって起動されるオペレータ・コントロール・モード、コンピュータ・プログラムによって起動されるプログラム・コントロール・モードを持つことを特徴とする請求項１３記載の情報コンテンツの連携管理システム。

【請求項１５】　コンテンツ・ファイル内のコンテンツ変数とそのコンテンツ・ファイルに対応するステーション内のポートとの間において、内部参照ポートが他のコンテンツ変数に関数依存節を通して依存するコンテンツ変数の集合に対応し、送信ポートが他のコンテンツ変数に依存しないコンテンツ変数であるフリー変数の集合に対応し、外部参照ポートが単独のフリー変数に対応し、ローカルポートがコンテンツ変数を持つ要素外にあるすべての要素に対応することを特徴とする請求項１２から請求項１４のいずれかに記載の情報コンテンツの連携管理システム。

【請求項１６】　ステーションのレイアウトが、ＸＳＬ(Extensible Styles heet Language)などのウェブ標準言語により、そのステーションと対応するコンテンツ・ファイルから分離して記述されることを特徴とする請求項１５記載の情報コンテンツの連携管理システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ウェブ上で情報コンテンツのネットワークを構築、管理することによって、地理的、機能的に分散された、人によるタスク（作業、業務、意志決定など）及びコンピュータ・プログラムによるタスクが効率よく協調、連動することをサポートする技術に係るものである。

【０００２】

【従来の技術】

ウェブ上で構築、閲覧される情報コンテンツは、ＨＴＭＬ、ＸＭＬなどのウェブ標準言語で記述される。これらの言語で記述された情報コンテンツの単位は一般にウェブ・ドキュメントと呼ばれている。ウェブ・ドキュメントの最も重要な特徴は、記述される情報コンテンツを階層木構造的に部品化することができ、しかも各部品コンテンツに対し、タグと呼ばれる形式を通して、その部品コンテンツの意味付け、属性などを表現できることである。ＨＴＭＬにおけるタグが情報コンテンツのレイアウトに関する意味付けで、既に定義され拡張性がないのに対し、ＸＭＬのタグはユーザによって一般的な意味付けが可能であり言語として拡張性を持つ。ＨＴＭＬやＸＭＬではこのようなタグによって部品化された情報コンテンツを「要素」と呼ぶ。

【０００３】

以下、図１を通して、ＸＭＬにおける要素及び要素間の階層構造を例示する。図１において、たとえば、＜会社名＞ＡＢＣ製紙＜／会社名＞の部分が要素の一例である。＜会社名＞が「開始タグ」と呼ばれ、＜／会社名＞が「終了タグ」と呼ばれる。要素はまた内部木構造を持つこともでき、要素の中に要素が存在することができる。この場合、後者の要素を前者の要素の「サブ要素」という。たとえば図１の例では、開始タグ＜勤務先＞から終了タグ＜／勤務先＞までの部分が内部木構造を持った要素の例である。またこの要素のサブ要素としては、＜会社名＞ＡＢＣ製紙＜／会社名＞、＜父会社連絡先　Ｃｎａｍｅ＝“Ｃ１”＞から＜／父会社連絡先＞までの部分などがある。後者のサブ要素には、さらにまたその

サブ要素としてとして、＜直通電話＞０２５　４３２　３２２９＜／直通電話＞などがある。ある要素のサブ要素のサブ要素もまたその要素のサブ要素であり、したがって、＜直通電話＞０２５　４３２　３２２９＜／直通電話＞もまた＜勤務先＞から始まる要素のサブ要素である。ＸＭＬではこれら要素に対し様々な属性を定義することができる。この場合、その要素の開始タグ内に、属性名＝“属性値”という形式で属性定義が表現される。図１の例では、「父情報」という要素が、その開始タグ

＜父情報　更新日＝“２月１日２０００年”＞

を通して「更新日」という属性を与えられ、その属性値が“２月１日２０００年”と定義されている。

【０００４】

ＨＴＭＬなど現在一般的に使われているウェブ標準言語では、ウェブ・ドキュメントの要素間にリンクをはることができ、このようなリンクを通してウェブ上に情報コンテンツのネットワークを実現できる。さらに、現在普及が進んでいるＸＭＬ及びその関連標準であるＸＬｉｎｋｓでは、これまでのＨＴＭＬと比べはるかに優れたリンク機能が可能となる。

【０００５】

しかしながら、リンクは、単にある情報コンテンツから他の情報コンテンツにアクセスするためのメカニズムであり、リンクで関連付けられた情報コンテンツ間に存在しうるコンテンツ上の「依存関係」をとらえるものではない。ここでいう「依存関係」とリンクとの決定的な相違は、前者にあってはその関連情報コンテンツ間に「解釈」が介在することである。たとえば、ある製品を開発したエンジニアは、その製品に関する販売情報に対してある特定の解釈を与える。したがって、その解釈に沿った形でこの販売情報が分析、加工され、その結果生成される情報が開発エンジニアに提示されることが望ましい。この場合、後者の情報コンテンツは前者の情報コンテンツに「依存」することになる。このような依存関係は、ＸＭＬ、ＸＬｉｎｋｓも含めたウェブ標準言語のリンクでは直接記述することはできない。

【０００６】

現在、このような依存関係を記述する方法としては、Ｊａｖａなどのプログラミング言語あるいはＪａｖａＳｃｒｉｐｔなどのスクリプト言語によるものが一般的である（以下、プログラミング言語及びスクリプト言語を総称してプログラミング言語と呼ぶ）。

【０００７】

しかし、プログラミング言語主体の依存関係記述には、以下のような難点、問題点がある。

・独立性の欠損：　依存関係の正確な定義内容が特定のプログラミング言語に依存することになり、依存関係記述の独立性が損なわれる。依存するプログラミング言語が変更されれば、既に構築された依存関係が無効になる可能性がある。

・共有性における難点：　依存関係の正確な定義内容を理解するためには、そのプログラミング言語の知識が必要であり、依存関係そのもの（すなわち、情報コンテンツの解釈）を知識として共有することが難しい。

・記述された依存関係の維持管理における柔軟性の問題：　プログラミング言語によって依存関係が一度構築されるとその調整、変更が難しく、硬直的な情報システムになりやすい。

・統一性の問題：　一般的な依存関係の記述方式が確立していないため、依存関係の記述における統一性が損なわれやすく、複数の依存関係を連動することが難しい。情報コンテンツＡ，Ｂ，Ｃがあるとする。ＡＢ間の依存関係とＢＣ間の依存関係に異なった記述方式が採用されると、複数の依存関係解釈実行環境が必要になり、ＡＣ間の連鎖的な依存関係の解釈実行が難しくなる。

・一貫性の問題：　統一的な依存関係の記述方式が確立していないと、依存関係を通してネットワーク化されら情報コンテンツ全体の一貫性を維持することが難しい。一部の情報コンテンツが更新された場合、その情報コンテンツに直接的、間接的に依存する全ての情報コンテンツが連鎖的に即座に更新されることが困難である。

【０００８】

次にウェブ上の情報コンテンツを更新したり、閲覧したりするためのインターフェースの問題に移る。情報送受信の中継点、あるいはインターフェースとして

の「ポート」という概念は古くからある。たとえば、電話送受信機がポートの例である。インターネットにおける電子メールもポートの例である。これらのポートは不特定の送信者より不特定の情報コンテンツが入ってくる、汎用性の強いポートである。これに対し、ウェブの出現によって、目的に沿って特化したポートの構築が可能になりつつある。たとえば、企業が用意した、顧客ごとにカスタマイズされたウェブページを通し、顧客が自分のニーズに合った情報を取得したり、あるいは自分の発注用件を送信したりする仕組みは、目的的に特化されたポートの初歩的な例と言える。

【０００９】

しかしながら、目的的ポートの目的性が特化すればするほど、そのポート使用者の取組むタスクにおけるそのポートの位置付けも特化する。タスク全体の遂行のためには複数のポートが必要となるであろうが、これらの関連ポートをまとめるフレームワークがない。たとえば、上記の製品開発の例で言えば、営業から入ってくる販売情報のポートも必要であろうし、製造現場から入ってくる製造過程に関する情報のポートも必要になる。また、製造現場に発信するためのポートも必要であろう。これら複数のポートをポート複合体として「製品開発」という業務の視点から関連付けるフレームワークが必要である。

【００１０】

現時点では、ウェブ上における目的的なデジタル・ポートの複合体はその概念そのものがまだ体系的に確立されておらず、技術上以下のような難点、問題点がある。

・ポート複合体レイアウトの非連続性：　たとえば、ウェブページを通して行われる発注タスクの場合、発注ポートと発注確認のための「確認ポート」は、発注というタスクの視点からは同一のポート複合体に含まれるべきであるが、たいてい発注ポートと確認ポートは異なるウェブ・ドキュメントとして表現され、発注タスクを担当するユーザは複数のウェブページを往来することになる。つまり、同一のポート複合体としてのレイアウト上及びポート操作上の連続性が損なわれている。同一ポート複合体におけるポート間のレイアウト上及びポート操作上の「距離」は、それらのポートで提示される情報量によってのみ決まるべきである

。

・ポート複合体構成要素の混在：　ポート複合体には基本的な構成要素がある。たとえば、ポート複合体で提示される情報コンテンツ、類別化されたポートの集合、各ポートの持つ入力あるいはコントロール機能、ポート複合体のレイアウトなどである。ＨＴＭＬを主体としてポート複合体を構築する場合、上記のすべての要素がＨＴＭＬファイルに混在することになり、ＨＴＭＬの知識をもった者がすべての要素を構築することになる。これではポート複合体の構築が効率的でない。これに対し、ＸＭＬは情報コンテンツとそのプレゼンテーションを分離するので、ポート複合体のレイアウトはたとえばＸＳＬなどを用いて別記することができる。しかしながら、ＸＭＬなどによる構築においても、上記構成要素が概念的に区別されていないため、ＨＴＭＬで行うポート複合体の構築作業を単にＸＭＬのレイアウトで繰り返すことになる。

【００１１】

【発明が解決しようとする課題】

本発明の一般的な課題は、地理的、機能的に分散された、人及びコンピュータによるタスクが、ウェブ上で適宜な情報コンテンツを適時に配信することによって、効率よく協調、連動するため、ウェブ上で情報コンテンツ及びその相互依存関係のネットワークを構築し、コンピュータによる情報コンテンツの連鎖更新を実現することである。本発明のより具体的な課題は、情報コンテンツ間の依存関係の記述において、その統一性、特定プログラミング言語からの独立性、知識としての依存関係の共有性、及び依存関係変更における柔軟性を高めることである。本発明のより具体的な他の課題は、相互に依存する情報コンテンツの連鎖更新を連続的に実行、管理することによって、関連する情報コンテンツ全体の一貫性を維持することである。本発明のより具体的な他の課題は、相互に依存する情報コンテンツの連鎖更新において、依存関係全体の依存構造を事前に分析することにより、連鎖更新の効率を向上することである。本発明の他の一般的な課題は、情報送受信の関連ポートの集合からなるポート複合体をウェブ上に構築することである。本発明のより具体的な課題は、ポート複合体における関連ポートが従来技術におけるようにそれぞれ別のウェブ・ドキュメントとして実現されることか

ら帰結する、ポート間の不必要な「距離」を排除することである。本発明のより具体的な他の課題は、ポート複合体の構成要素を分離定義し、ポート複合体構築の効率を向上することである。本発明のより具体的な他の課題は、ウェブ上に構築された情報コンテンツ及びその相互依存関係のネットワーク上にポート複合体のネットワークを実現することである。

【００１２】

【課題を解決するための手段】

本発明では、ウェブ・ドキュメントの集合として表現された情報コンテンツにおいて、その情報コンテンツ間の依存関係をウェブ・ドキュメントの要素間の依存関係としてとらえる。このような依存関係はプログラミング言語で記述することも可能であるが、本発明では、より好ましい形態として、ＸＭＬ及びＸＭＬで定義された言語であるＭａｔｈＭＬなどのウェブ標準言語で記述する。このことによって、依存関係の定義がその解釈実行システムから分離、独立する。さらに、依存関係の定義そのものがＸＭＬ及びＭａｔｈＭＬによるウェブ・ドキュメントあるいはその集合として表現されるので、依存関係全体の依存構造の分析が容易になる。したがって、全体依存構造を事前に分析することによって、より効率のよい依存関係の解釈実行システムの構築が可能となる。また、ウェブ・ドキュメント内の任意の要素が他のウェブ・ドキュメント内の要素に対し、コンピュータによる解釈実行が可能な依存関係を持つことによって、ウェブ・ドキュメント全体ではなくその一部である任意の要素がウェブ上における情報送受信のポートとして機能することが可能になる。したがって、関連する複数のポートを同一のウェブ・ドキュメント内に設けることにより、ユーザが関連ポート間を移動する際、複数のウェブ・ドキュメントをダウンロードし、それらのドキュメント間を往来する必要がなくなる。要素間の依存関係を持つウェブ・ドキュメントの集合においてそれぞれのウェブ・ドキュメントに、ポート複合体としてのレイアウトを与えることによって、互いに連携するポート複合体のネットワークがウェブ上に実現する。また、この場合、ポート複合体の構成要素、つまり、ポート複合体で提示される情報コンテンツ、類別化されたポートの集合、各ポートの持つ入力あるいはコントロール機能、ポート複合体のレイアウトを分離することによりポ

ート複合体構築の効率をあげることができる。

【００１３】

【発明の実施の形態】

要素間の依存関係を伴ったウェブ・ドキュメントの集合をコンテンツ・ネットと呼ぶ。コンテンツ・ネットにおける要素間の依存関係は、コンピュータによる解釈実行が可能な形式が与えられているものとする。

【００１４】

コンテンツ・ネットにおいて、依存関係の対象となる要素をアクティブ要素と呼ぶ。他のアクティブ要素に依存しないアクティブ要素をイニシアル要素と呼ぶ。イニシアル要素のコンテンツは、コンテンツ・ネットのユーザ、あるいはコンテンツ・ネットの外部にあるデータベースなどのアプリケーション・プログラムによって、入力または更新される。１あるいはそれ以上のイニシアル要素にコンテンツ変更が生じた場合、それらのイニシアル要素のいずれかに直接的、間接的に依存する全てのアクティブ要素を順次、自動的に連鎖更新するシステムを「コンテンツ・ネット連鎖更新システム」と呼ぶ。

【００１５】

コンテンツ・ネットにおけるアクティブ要素は、互いに識別可能な独自の形式（識別子と呼ぶ）が与えられているものとする。識別子の形式としては、たとえば、ウェブ・ドキュメントの木構造に基づき、そのウェブ・ドキュメントにおける最上位のルート要素から識別の対象となっているアクティブ要素までのパスがあげられる。このようなパスを表現する言語としては、ＸＰａｔｈあるいはＸＰｏｉｎｔｅｒｓなどのウェブ標準化が進んでいる言語がある。識別子の他の形式としては、各アクティブ要素に与えられた特定属性の属性値を通して表現することも可能である。実施形態の一例として、ここでは「Ｃｎａｍｅ」という属性による後者の方式でアクティブ要素を識別する。Ｃｎａｍｅ属性の属性値をコンテンツ変数と呼ぶ。図１の例では、「父会社連絡先」という要素が、その開始タグ

＜父会社連絡先　Ｃｎａｍｅ＝“Ｃ１”＞

を通して、“Ｃ１”というコンテンツ変数を与えられている。

コンテンツ・ネットにおける依存関係は、このようなコンテンツ変数などの識

別子を通して記述される。

【００１６】

図９では、識別子を通して記述された依存関係を持つコンテンツ・ネットの１実施形態を例示する。この例では、ａ．ｘ，ａ．ｙなどがアクティブ要素に対応する識別子であり、アクティブ要素間の依存関係が識別子を通してF_1からF_8までの関数で定義されている。図９における制約ファイルは、このコンテンツ・ネットにおけるすべての依存関係を含んでいるが、これはコンテンツ・ネットの１実施形態であって、他の実施形態としては依存関係の記述をいくつかの制約ファイルに分散することも可能である。たとえば、各コンテンツ・ファイルと対になった制約ファイルでそのコンテンツ・ファイル内のアクティブ要素の依存関係を定義するという形態も可能である。図１０は、このようなコンテンツ・ネットを例示している。さらに、他の実施形態として、コンテンツ・ファイルと制約ファイルを分離せず、制約ファイル内の依存関係をコンテンツ・ファイルのウェブ・ドキュメント内に取り込むことも可能である。特に情報コンテンツを表すウェブ・ドキュメントがＸＭＬで記述されている場合、あるいはＸＭＬによるタグ付けを部分的に許す場合、依存関係をコンテンツ・ファイルに取り込むことは容易になされる。

【００１７】

図９における制約ファイルは、アクティブ要素の連鎖更新を実現するにあたって、具体的にはＪａｖａなどのプログラミング言語で実装したものでもよいし、あるいはこの制約ファイルにある構文を解釈実行するシステムを構築することも可能である。F_1などの関数は上記プログラミング言語や依存関係解釈実行システム内で処理されるものであってもよいし、あるいは以下の条件を満たす限りにおいて外部のアプリケーション・ソフトウエアで実現されることも可能である。

・上記アプリケーション・ソフトウエアとウェブ・ドキュメントの間にデータの読出し、書込みのインターフェースを構築できる。

・上記読出し、書込みの操作及び上記アプリケーション・ソフトウエアの起動がその外部にあるプログラムでコントロールできる。

F_1などの関数がこのようなアプリケーション・ソフトウエアを通して実現さ

れる場合、コンテンツ・ネット連鎖更新システムは、様々のアプリケーション・ソフトウエアをウェブ・ドキュメントを介して連動するシステムとなる。

【００１８】

以下、コンテンツ・ネット連鎖更新システムについてさらに詳細に説明する。以下の説明では、アクティブ要素の識別子をコンテンツ変数で表現し、コンテンツ・ネットをコンテンツ・ファイルと制約ファイルの対の集合として解説するが、これは実施の１形態に過ぎない。

【００１９】

コンテンツ変数Ｃがコンテンツ・ファイルＸのあるアクティブ要素に与えられているとする。この時、「ＣがＸで宣言されている」といい、そのアクティブ要素を「Ｃ要素」と呼ぶ。アクティブ要素は他のアクティブ要素をサブ要素として持つことができる。アクティブ要素Ｂがアクティブ要素Ａのサブ要素であるとする。ＡとＢの間に他のアクティブ要素（つまり、Ｂをサブ要素とするアクティブ・サブ要素）が存在しない場合、ＢはＡのダイレクト・アクティブ・サブ要素と言う。

【００２０】

コンテンツ・ネットにおけるコンテンツ変数間の依存関係は、「依存節」で定義される。依存節は以下の形式、あるいはそれと同等の内容を持つ他の形式で表現される。

$[C_0, \{C_1, C_2, \cdots, C_n\}, \quad d]$,

上記におけるC_0, C_1, C_2, …, C_nはすべてコンテンツ変数である。このような依存節の意味は、コンテンツ変数C_0が他のコンテンツ変数C_1, C_2, …, C_nによって依存節の第３項の“ｄ”を通して定義されているという意味である。コンテンツ変数C_0が定義されている依存節を「C_0依存節」と呼ぶ。コンテンツ・ネットでは各コンテンツ変数に対し、その変数を定義する依存節が存在する場合にはその依存節が１つしかないことを想定する。依存節を持たないコンテンツ変数を「フリー変数」と呼ぶ。従って、フリー変数は、他のアクティブ要素に依存しないイニシアル要素と対応し、ユーザあるいはコンテンツ・ネット外部のアプリケーションからの入力インターフェースとして機能する。

【００２１】

依存節には「関数依存節」と「階層依存節」がある。関数依存節は以下の条件を満たす。

・依存節の第３項の“ｄ”はコンテンツ変数C_1，C_2，…，C_nとＭａｔｈＭＬで提供される演算子、ＭａｔｈＭＬで参照されうる外部演算子が関数的に構成された表現である。

・依存節の第１項及び第２項における、異なるコンテンツ変数は異なるコンテンツ・ファイルで宣言されていてもよい。ただし、第２項のコンテンツ変数C_iに対し、C_i要素とC_0要素にはサブ要素関係はない。

これに対し、階層依存節は以下の条件を満たす。

・依存節の第３項の“ｄ”は何もしないという意味のＮｕｌｌ演算を表し、たとえばシンボル“ｎｕｌｌ”でこれが示されている。

・C_0が宣言されているコンテンツ・ファイルが存在し、このC_0要素がC_1要素、C_2要素、…C_n要素をそのすべてのダイレクト・アクティブ・サブ要素とする。

【００２２】

関数依存節の第１項にあるコンテンツ変数は「関数依存変数」、あるいは単に「依存変数」と呼ばれる。階層依存節の第１項にあるコンテンツ変数は「集成変数」と呼ばれる。コンテンツ変数Ｃが依存節を持つとする。Ｃ依存節は１つしかないので、コンテンツ変数Ｃは依存変数か集成変数のどちらかであり、同時に両者であることはない。また、Ｃ要素が他のアクティブ要素をサブ要素とする場合、Ｃは階層依存節で定義され集成変数となる。したがって、依存変数を持つ要素は他のアクティブ要素をサブ要素として持つことはない。

【００２３】

階層依存節はコンテンツ・ファイルにおけるアクティブ要素間の階層依存で決定される。これに対し、関数依存節は直接制約ファイルで記述される。関数依存を例をあげて説明するため、さらに図２、図３、図４にある３つのコンテンツ・ファイルを想定する。図１、図２、図３、図４にある４つのコンテンツ・ファイルに対し以下の関数依存節を考える。

[Z，{Ｃ２，Ｈ２}，Ｃ２＋Ｈ２]

[W，{Ｃ１，Ｈ１}，buildTree（Ｃ１，Ｈ１）]

依存節[Z，{Ｃ２，Ｈ２}，Ｃ２＋Ｈ２]は「佐藤家家計」コンテンツ・ファイルと対になる制約ファイルで記述され（図６）、依存節[W，{Ｃ１，Ｈ１}，buildTree（Ｃ１，Ｈ１）]は「佐藤家電話連絡簿」コンテンツ・ファイルと対になる制約ファイル（図８）で記述される。

【００２４】

依存節[Z，{Ｃ２，Ｈ２}，Ｃ２＋Ｈ２]の解釈は、Ｃ２で表現されるコンテンツ“１，０００，０００”とＨ２で表現されるコンテンツ“８００，０００”が合計され、その合計値がＺのコンテンツとなるという意味である。したがって、この依存節の解釈実行後、「佐藤家家計」コンテンツ・ファイルは図５のようになる。依存節の解釈実行の仕組みは後述する。

【００２５】

２番目の依存節[W，{Ｃ１，Ｈ１}，buildTree（Ｃ１，Ｈ１）]における「buildTree」はＭａｔｈＭＬで参照される外部演算子の１つで、その意味はＣ１で表現される木構造とＨ１で表現される木構造を隣り合わせに並べ組み、新たな木構造を生成することである。したがって、この依存節の解釈後、「佐藤家電話連絡簿」コンテンツ・ファイルは図７のようになる。

【００２６】

コンテンツ・ネットに対し、その制約ファイルで定義されるすべての関数依存節の集合、そしてそのコンテンツ・ファイルにおけるアクティブ要素間の階層依存によって決まるすべての階層依存節の集合の和集合を、そのコンテンツ・ネットの依存グラフと呼ぶ。与えられたコンテンツ・ネットに対し、その制約ファイル及びコンテンツ・ファイルから全ての関数依存節及び階層依存節を抽出し、依存グラフを構築するモジュールを依存グラフ構築モジュールと呼ぶ（図１３参照）。この依存グラフにおいて、コンテンツ変数Ａがコンテンツ変数Ｂに依存し、さらにＢがコンテンツ変数Ｃに依存するとき、「ＡはＣに間接依存する」あるいは単に「ＡはＣに依存する」と言う。依存グラフから、各コンテンツ変数に対し、そのコンテンツ変数に直接的、間接的に依存する全てのコンテンツ変数を検出

したものを依存マトリクスと呼ぶ。具体的には、与えられたコンテンツ・ネットにおいて、コンテンツ変数の総数がｎ個の場合、依存マトリクスはコンテンツ変数をインデクスとするｎ×ｎのマトリクスとなる。依存マトリクスでは、コンテンツ変数C_1とC_2によって決まるマトリクス・エントリがC_1、C_2間の依存関係の有無を表す。このような依存マトリクスは公知の方法によって依存グラフから計算することができる。依存グラフから依存マトリクスを計算するモジュールを依存マトリクス構築モジュールと呼ぶ（図１３参照）。依存マトリクスによって、自己自身に間接依存するコンテンツ変数の依存サイクルの有無が検知される。このような依存サイクルがあると、依存関係に従ってアクティブ要素を連鎖更新するシステムは無限ループに陥ってしまう。依存マトリクスから、自己依存サイクルの有無を検知、レポートするモジュールを自己依存サイクル判定モジュールと呼ぶ（図１３参照）。フリー変数の集合に対し、その集合内のいずれかの変数に依存する全てのコンテンツ変数の集合を更新候補変数集合と言う。与えられたフリー変数集合の更新候補変数集合を依存マトリクスより算出するモジュールを更新候補変数集合決定モジュールと呼ぶ。フリー変数は更新スコープごとに同時に更新される変数の集合としてまとまっている。たとえば、発注タスクにおいて、発注品目、発注量、発注日などがそれぞれ個々のコンテンツ変数を持つ場合、このようなコンテンツ変数は一組となって情報更新の単位を表す。このような更新スコープが予め定まっている場合は、そのスコープ内のコンテンツ変数集合に対する更新候補変数集合を事前に算出しておくことができる。上記４つのモジュール、すなわち、依存グラフ構築モジュール、依存マトリクス構築モジュール、自己依存サイクル判定モジュール及び更新候補変数集合決定モジュールからなるモジュールを依存構造分析モジュールと呼ぶ。

【００２７】

コンテンツ・ネットにおけるユーザからの更新リクエストは、フリー変数の集合と、各フリー変数と対になる更新コンテンツの集合からなる。与えられた更新リクエストのフリー変数の集合を受けて、前記依存構造分析モジュールがそのフリー変数の集合に対する更新候補変数集合を決定する。更新候補変数集合内のコンテンツ変数に対応するアクティブ要素を順次更新していくモジュールを更新モ

ジュールと呼ぶ。更新モジュールの実施形態には様々なものがあるが、大別してトップダウン・モジュールとボトムアップ・モジュールがある。トップダウン・モジュールは公知の方式に基づくもので、選ばれた更新候補変数からスタートし、再帰的に更新可能なコンテンツ変数をサーチし、そのようなコンテンツ変数を検知すると対応するアクティブ要素を更新していく。ここでいう更新可能なコンテンツ変数（Ｃとする）とは、更新されたコンテンツ変数集合（Ｓとする）に対する相対的なもので、以下の条件を満たすコンテンツ変数を言う。

ＣはＳに属する少なくても１つの変数に直接的に依存する。

ＣがＳに属さないあるコンテンツ変数、Ｄ、に直接的に依存するとき、ＤはＳ内のどの変数にも依存しない（したがってＤはＳに影響されることはない）。

【００２８】

このようなトップダウン・モジュールは実装も容易で、依存連鎖のチェーンが短いときは有効である。依存連鎖のチェーンが長いときは、更新可能なコンテンツ変数を予め計算しておくことが有効である。この場合、まず与えられた更新リクエストのフリー変数の集合に対するすべての更新可能なコンテンツ変数を判明する。与えられた更新リクエストのフリー変数には更新順位を表す更新ランク０を与え、判明された更新可能なコンテンツ変数には更新ランク１を与える。さらに、更新ランク１を与えられた全てのコンテンツ変数の集合に対し、更新可能なコンテンツ変数を判明し、それらのコンテンツ変数に更新ランク２を与える。このような手続きを繰り返すことによって、与えられた更新リクエストのフリー変数の集合に対するすべての更新候補変数に更新ランクを与えるモジュールを更新ランクモジュールと呼ぶ。更新ランクモジュールは依存マトリクスを使うことによって、更新ランクを効率よく計算できる。図９における例でいうと、以下の更新ランクが更新ランクモジュールによって決定される。

更新ランク０：　ａ．ｘ，ｃ．ｙ，ｄ．ｚ

更新ランク１：　ｄ．ｘ，ｄ．ｙ

更新ランク２：　ｂ．ｙ，ｃ．ｘ

更新ランク３：　ｂ．ｚ

更新ランク４：　ａ．ｚ

更新ランク５：　ａ．ｙ

更新ランク６：　ｂ．ｘ

【００２９】

なお、図９では、このような更新ランクは識別子のとなりに示されている。前記ボトムアップ・モジュールの一形態として、このような更新ランクモジュールを用い、更新ランクの低いコンテンツ変数から順にアクティブ要素を更新していく更新モジュールが有効である。また、同一の更新ランクを持つコンテンツ変数間には依存関係がないので、同一の更新ランクを持つコンテンツ変数に対応するアクティブ要素の更新は並列的に処理ができさらに効率がよい。

【００３０】

トップダウン・モジュール及びボトムアップ・モジュールにおいて、更新可能なコンテンツ変数が判明するにしたがって、そのコンテンツ変数が定義されている依存節に基づいて、対応するアクティブ要素を順次更新していくモジュールを連鎖的依存関係解釈実行モジュールと呼ぶ。したがって、ボトムアップ・モジュールが更新ランクモジュールと連鎖的依存関係解釈実行モジュールからなるのに対し、トップダウン・モジュールは再帰的な方法により更新可能なコンテンツ変数を判明するモジュールと連鎖的依存関係解釈実行モジュールからなる。図１３は、ボトムアップ・モジュールを持つコンテンツ・ネット連鎖更新システムの一形態を示すが、図中の更新ランク・モジュールを、コンテンツ変数を再帰的に判明するモジュールに置換え、さらにランク付更新候補変数集合を更新可能な変数に置き換えれば、トップダウン・モジュールを更新モジュールとする、コンテンツ・ネット連鎖更新システムの他一形態を示すことになる。

【００３１】

コンテンツ・ネットにおける各コンテンツ・ファイルに対し、そのレイアウトを定義する「スタイル・ファイル」を想定する。スタイル・ファイルがその対象となるコンテンツ・ファイルとは別ファイルになっていることを想定しているが、これは一実施形態であり、ＨＴＭＬなどによりスタイル・ファイルとコンテンツ・ファイルが混在している形態も可能である。コンテンツ・ファイルがスタイル・ファイルを通して、あるいはＨＴＭＬなどにより直接的にレイアウトされた

ものを「プロセス・ステーション」あるいは単に「ステーション」と呼び、コンテンツ・ネットによりネットワーク化されたステーションの集合を「ステーション・ネット」と呼ぶ（図１２参照）。ただし、この場合のレイアウトはプロセス・ステーションの構造及び機能を反映したものでなければならない。ここでは、プロセス・ステーションの構造及び機能を説明し、その後コンテンツ・ファイルの各要素がプロセス・ステーションの構造及び機能にどのようにマッピングするか解説する。

【００３２】

ステーションは構造的には１あるいはそれ以上の「ポート」からなる。ポートには少なくても以下のタイプがある（図１１）。

参照ポート

・内部参照ポート：　ネット内部情報を、関数依存を通して参照する。

・外部参照ポート：　ネット外部情報を取入れ参照する。

インプットポート

・送信ポート：　ステーションにおける入力インターフェースで、入力されたコンテンツはネット内部情報の更新連鎖を起動する。

・ローカルポート：　ステーション固有のメモスペースで、入力された情報はステーション内に留まる。

【００３３】

各ポートはいくつかの「コントロール機能」を持つことができる。コントロール機能には基本機能といくつかの基本機能の組合せで決まる複合機能とがある。基本機能にはすくなくとも以下のものがある。

・内部更新：　ネット内部情報を依存節で定義された依存関係に従って更新していく。

・インポート：　ネット外部情報をネット内部に取り入れる。

・エクスポート：　ネット内部情報をネット外部のアプリケーションに送る。

【００３４】

ステーションでは、基本機能から複合機能を定義するためにいくつかの「コントロール・アセンブラー」が用意されている。コントロール・アセンブラーには

少なくても以下のものがある。

逐次実行：シンボル“｜”で表す。

同時実行：シンボル“＆”で表す。

【００３５】

したがって、複合機能には、たとえば以下のものがある。

・インポート｜内部更新

インポート実行後、内部更新実行

・内部更新＆エクスポート

内部更新とエクスポートを同時実行

・インポート｜（内部更新＆エクスポート）

インポート実行の後、内部更新及びエクスポートを同時実行

【００３６】

さらに、各コントロール機能には、基本機能であれ複合機能であれ、「コントロール・モード」が割当てられる。「コントロール・モード」には少なくとも以下のものがある。

・オペレータ・コントロール：　対象となるコントロール機能がステーションのユーザによって起動される。

・プログラム・コントロール：　対象となるコントロール機能がコンピュータ・プログラムによって起動される。

【００３７】

コンテンツ・ファイルにおいて、依存変数を持った要素を内部参照要素と呼ぶ。内部参照要素は更新スコープごとにグループ化し、同一の更新スコープに属する内部参照要素の集合がステーションの内部参照ポートに対応する。ただし、このような内部参照要素の集合は１個の内部参照要素からなることもある。フリー変数を持った要素にはイニシアル要素と外部参照要素の２つのタイプがある。イニシアル要素は更新スコープごとにグループ化し、同一の更新スコープに属するイニシアル要素の集合がステーションの送信ポートに対応する。ただし、このようなイニシアル要素の集合は１個のイニシアル要素からなることもある。外部参照要素はステーションの外部参照ポートに対応する。コンテンツ変数を持たず、

アクティブ要素のサブ要素となっていないすべての要素の集合がステーションのローカルポートに対応する。

【００３８】

各コンテンツ・ファイルに対し、上記のマッピングに準拠しながら、スタイル・ファイルが作成される。スタイル・ファイルの記述にはＸＳＬなどの世界標準もあり、またＸＳＬなどをサポートするブラウザもある。このようなブラウザにコンテンツ・ファイルとそれに対応するスタイル・ファイルを入力すれば、ステーションがウェブ上で作動する。

【００３９】

上記のようにコンテンツ・ファイルとスタイル・ファイルが別ファイルとして存在する形態でも、あるいはＨＴＭＬなどを通して両者が混在する形態でも、前記コンテンツ・ネット連鎖更新システム上に、ポート複合体としてのステーションのネットワークを構築できる。このようにネットワーク化されたステーションは以下の代表的な特徴を持つ。

・関連する複数のポートが同一のウェブ・ドキュメント内に実現することにより、ユーザが関連ポート間を移動する際、複数のウェブ・ドキュメントをダウンロードし、それらのドキュメント間を往来する必要がなくなる。また、複数ポートのレイアウトが１個のウェブ・ドキュメントのレイアウト設定で決まるので、発注ポートと発注に対する返答ポートなど、コンテンツ的あるいはタスク的に関連の強いポートを隣接させることが容易である。

・ウェブ環境があればどこにでも設置できる複数のステーションをまたがって、様々なポートで提示される情報コンテンツの依存関係が連鎖更新されるので、人及びコンピュータ・プログラムによるタスクで必要とされる情報コンテンツが適宜な構成でしかもリアル・タイムで配信でき、これらタスクのすばやい連動、協調を可能とする。

【００４０】

典型的には、このようなネットワークは複数のステーションを含み、複数の人による複数のタスクの連携をサポートする。ただし、最も単純なケースとして、１個のステーションしか持たないステーション・ネットの形態も可能であり、こ

れまでのステーション・ネットの定義、解説からはずれるものではない。たとえば、前記発注タスクの例で言えば、このタスクに対応した「発注ステーション」のみからなるステーション・ネットも可能である。この場合、発注ステーションの送信ポートとしては、クレジット・カード情報を入力するポート、発注内容を入力するポート、商品カタログを閲覧するためのサーチ入力ポートなどが考えられる。また、これらの送信ポートにそれぞれ対応した内部参照ポートとしては、クレジット・カード情報を自動処理して適宜な返答を提示するポート、発注受理をコンピュータ・プログラムによって判断しその結果を提示するポート、コンピュータによるカタログ・サーチ結果を提示するポートなどがあげられる。この場合、これらの内部参照ポートと送信ポートの情報コンテンツ上の関係は、それらのポートと対応するコンテンツ変数間の依存節として記述することができる。たとえば、クレジット・カード情報を自動処理するプログラムをＭａｔｈＭＬのユーザ定義による外部関数として実現すれば、依存節の第３項はこの外部変数で表現される。クレジット・カード情報の入力ポートとその返答ポートを隣接させることによって、発注ステーションのユーザは同一画面上でクレジット・カード受理の確認をできる。

【図面の簡単な説明】

【図１】

本発明におけるコンテンツ変数が宣言されているコンテンツ・ファイルの例を示す図である。

【図２】

本発明におけるコンテンツ変数が宣言されているコンテンツ・ファイルの例を示す図である。

【図３】

本発明における関数依存変数が宣言されているコンテンツ・ファイルの例を示す図である。

【図４】

本発明における関数依存変数が宣言されているコンテンツ・ファイルの例を示す図である。

【図５】

本発明における宣言されている関数依存変数の定義が解釈実行された後のコンテンツ・ファイルの例を示す図である。

【図６】

本発明の制約ファイルの例を示す図である。

【図７】

本発明における宣言されている関数依存変数の定義が解釈実行された後のコンテンツ・ファイルの例を示す図である。

【図８】

本発明の制約ファイルの例を示す図である。

【図９】

本発明における要素間の依存関係が、情報コンテンツを表すコンテンツ・ファイルの集合とは別に単一の制約ファイルとして記述されているコンテンツ・ネットの例を示す図である。

【図１０】

本発明におけるコンテンツ・ファイルと制約ファイルが対となっているコンテンツ・ネットの例を示す例である。

【図１１】

本発明のステーションの例を示す図である。

【図１２】

本発明のステーション・ネットの一形態を例示する図である。

【図１３】

本発明のコンテンツ・ネット連鎖更新システムの一形態を示す図である。

【書類名】　　図面

【図１】

「父情報」コンテンツ・ファイル

```
＜?xml version="1.0" standalone="yes"＞
＜!--　ファイル名：　父情報.xml--＞
＜父情報　更新日＝“2 月 1 日 2000 年”＞
        ＜名前＞
        佐藤洋一郎
        ＜/名前＞
        ＜勤務先＞
                ＜会社名＞
                ABC 製紙
                ＜/会社名＞
                ＜会社住所＞
                新潟市内幸町３－５
                ＜/会社住所＞
                ＜会社電話番号＞
                ０２５　４３２　３２２１
                ＜/会社電話番号＞
                ＜父会社連絡先　Cname＝“C1”＞
                        ＜直通電話＞
                        ０２５　４３２　３２２９
                        ＜/直通電話＞
                        ＜秘書＞
                                ＜秘書名＞
                                森田健
                                ＜/秘書名＞
                                ＜秘書電話＞
                                ０２５　４３２　３２３１
                                ＜/秘書電話＞
                        ＜/秘書＞
                ＜/父会社連絡先＞
        ＜/勤務先＞
        ＜年収　Cname=“C2”＞
        1,000,000
        ＜/年収＞
＜/父情報＞
```

【図２】

「母情報」コンテンツ・ファイル

```
<?xml version="1.0" standalone="yes">
<!--  ファイル名 :  母情報.xml-->
<母情報>
        <名前>
        佐藤恵子
        </名前>
        <勤務先>
                <会社名>
                いろは製菓
                </会社名>
                <会社住所>
                新潟市寺尾町１２－８
                </会社住所>
                <会社電話番号>
                ０２５　８７６　１４５３
                </会社電話番号>
                <母会社連絡先　Cname＝“H1”>
                        <直通電話>
                        ０２５　８７６　１４５８
                        </直通電話>
                        <非常時電話>
                        ０２５　８７６　１４５０
                        </非常時電話>
                </母会社連絡先>
        </勤務先>
        <年収　Cname＝“H2”>
        800,000
        </年収>
</母情報>
```

【図３】

「佐藤家家計」コンテンツ・ファイル

```
<?xml version="1.0" standalone="yes">
<!--  ファイル名 :  佐藤家家計.xml  -->
<佐藤家家計>
        <年収入　Cname＝“Z”>
        </年収入>
        <年支出>
                <食費>
                </食費>
                <住宅ローン>
                </住宅ローン>
                <その他>
                </その他>
        </年支出>
</佐藤家家計>
```

【図４】

「佐藤家電話連絡簿」コンテンツ・ファイル

```
＜?xml version="1.0" standalone="yes"＞
＜!--　ファイル名：　佐藤家電話連絡簿.xml　--＞
＜佐藤家電話連絡簿＞
        ＜自宅＞
        ０２５　６３４　９１２０
        ＜/自宅＞
        ＜勤務先　Cname＝“W”＞
        ＜/勤務先＞
＜/佐藤家電話連絡簿＞
```

【図５】

「佐藤家家計」コンテンツ・ファイル：

依存節[Z, (C2, H2), C2+H2]の解釈実行後

```
＜?xml version="1.0" standalone="yes"＞
＜!--　ファイル名：　佐藤家家計.xml　--＞
＜佐藤家家計＞
        ＜年収入　Cname＝“Z”＞
        1,800,000
        ＜/年収入＞
        ＜年支出＞
                ＜食費＞
                ＜/食費＞
                ＜住宅ローン＞
                ＜/住宅ローン＞
                ＜その他＞
                ＜/その他＞
        ＜/年支出＞
＜/佐藤家家計＞
```

【図６】

「佐藤家家計」制約ファイル

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Constraints SYSTEM "http://www.iuj.ac.jp/xml/Constraints.dtd">
<Constraints>
        <Target docURL="http://www.iuj.ac.jp/xml/佐藤家家計.xml">
        佐藤家家計
        </Target>
        <Reference docURL="http://www.iuj.ac.jp/xml/父情報.xml">
        父情報
        </Reference>
        <Reference docURL="http://www.iuj.ac.jp/xml/母情報.xml">
        母情報
        </Reference>
        <Definition>
                <Cvariable >    Z   </Cvariable>


                <math>
                        <apply>
                                <plus/>
                                <ci> C2 </ci>
                                <ci> H2 </ci>
                        </apply>
                </math>
        </Definition>
</Constraints>
```

【図７】

「佐藤家電話連絡簿」コンテンツ・ファイル：

依存節[W, (C1, H1), buildTree(C1, H1)]の解釈実行後

```
<?xml version="1.0" standalone="yes">
<!--  ファイル名：  佐藤家電話連絡簿.xml  -->
<佐藤家電話連絡簿>
        <自宅>
        ０２５  ６３４  ９１２０
        </自宅>
        <勤務先  Cname＝“W” >
                <父会社連絡先>
                        <直通電話>
                        ０２５  ４３２  ３２２９
                        </直通電話>
                        <秘書>
                                <名前>
                                森田健
                                </名前>
                                <電話>
                                ０２５  ４３２  ３２３１
                                </電話>
                        </秘書>
                </父会社連絡先>
                <母会社連絡先>
                        <直通電話>
                        ０２５  ８７６  １４５８
                        </直通電話>
                        <非常時電話>
                        ０２５  ８７６  １４５０
                        </非常時電話>
                </母会社連絡先>
        </勤務先>
</佐藤家電話連絡簿>
```

【図８】

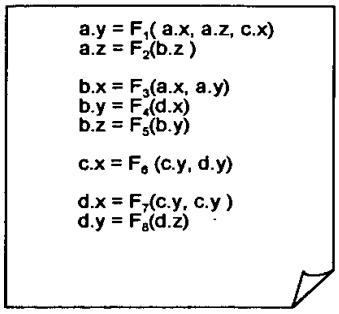
「佐藤家電話連絡簿」制約ファイル

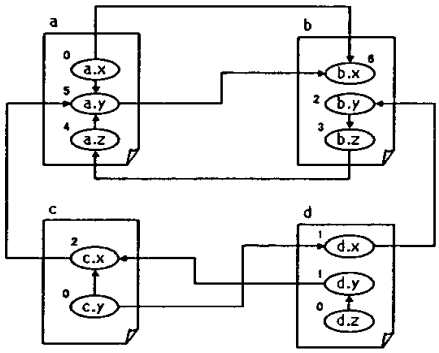
```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Constraints SYSTEM "http://www.iuj.ac.jp/xml/Constraints.dtd">
<Constraints>
        <Target docURL="http://www.iuj.ac.jp/xml/佐藤家電話連絡簿.xml">
        佐藤家電話連絡簿
        </Target>
        <Reference docURL="http://www.iuj.ac.jp/xml/父情報.xml">
        父情報
        </Reference>
        <Reference docURL="http://www.iuj.ac.jp/xml/母情報.xml">
        母情報
        </Reference>
        <Definition>
                <Cvariable >    W   </Cvariable>
                <math>
                        <apply>
                                <buildTree/>
                                <ci> C1 </ci>
                                <ci> H1 </ci>
                        </apply>
                </math>
        </Definition>
</Constraints>
```

【図９】

コンテンツ・ネットの例

- コンテンツ・ファイルの集合： {a, b, c, d}
- 依存関係が別ファイルとして記述された制約ファイル





注記：

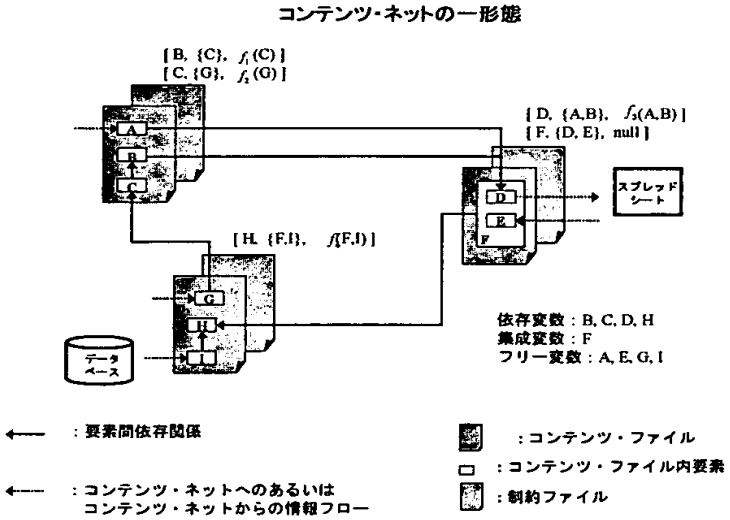
□ ： ウェブ・ドキュメント

○ ： アクティブ要素

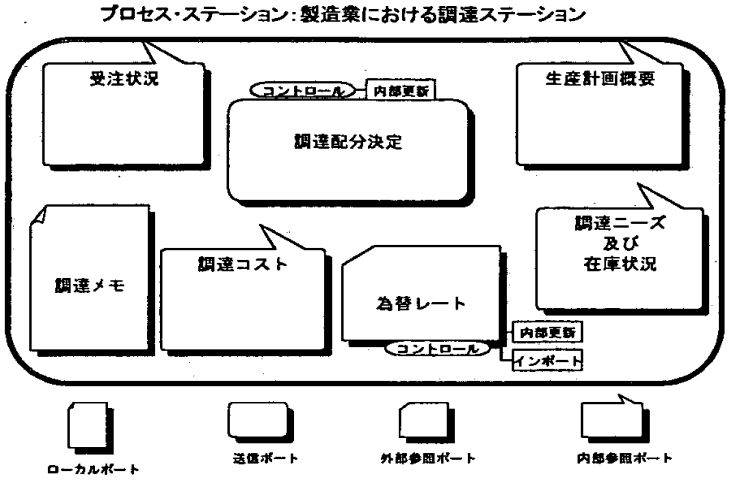
a.x, a.y, ... d.y, d.z ： 識別子

a.x, c.y, d.z ： イニシアル要素

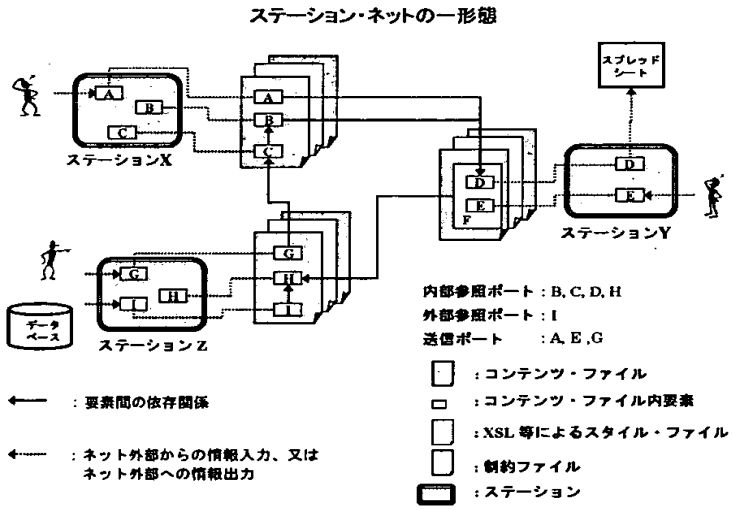
【図１０】



【図１１】

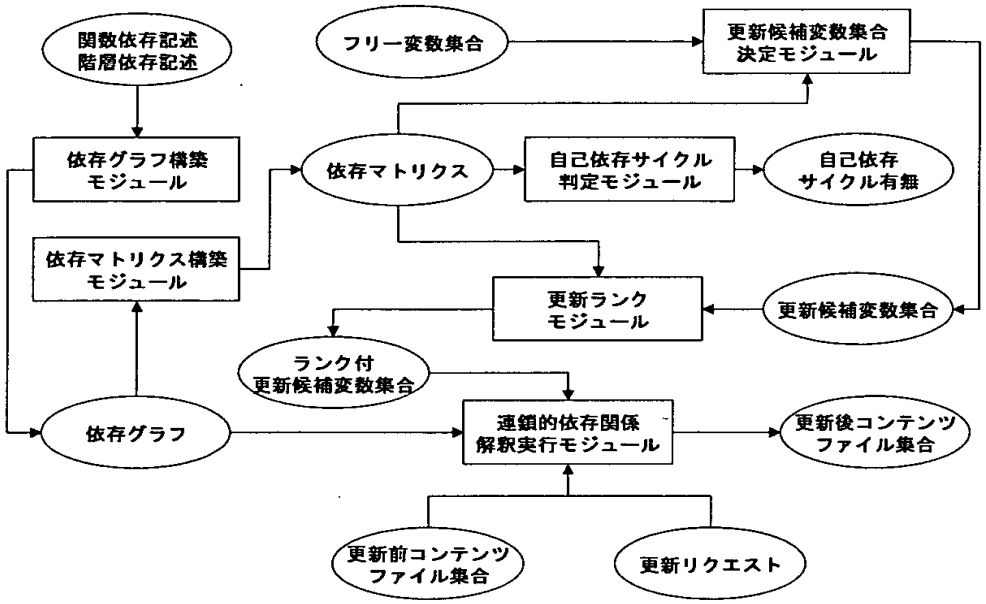

プロセス・ステーション：製造業における調達ステーション
受注状況
コントロール
内部更新
調達配分決定
生産計画概要
調達メモ
調達コスト
為替レート
調達ニーズ
及び
在庫状況
内部更新
コントロール
インポート
ローカルポート
送信ポート
外部参照ポート
内部参照ポート

【図１２】


ステーション・ネットの一形態
スプレッド
シート
ステーションX
ステーションY
ステーションZ
データ
ベース
内部参照ポート：B, C, D, H
外部参照ポート：I
送信ポート　：A, E, G
：コンテンツ・ファイル
：コンテンツ・ファイル内要素
：XSL 等によるスタイル・ファイル
：制約ファイル
：ステーション
：要素間の依存関係
：ネット外部からの情報入力、又は
ネット外部への情報出力

【図１３】

コンテンツ・ネット連鎖更新システムの一形態



【書類名】　　要約書

【要約】

【課題】　地理的、機能的に分散された、人及びコンピュータによるタスクが、ウェブ上で適宜な情報コンテンツを適時に配信することによって、効率よく協調、連動するため、ウェブ上で情報コンテンツ及びその相互依存関係のネットワークを構築し、コンピュータによる情報コンテンツの連鎖更新を実現するシステムを提供する。

【解決手段】　ウェブ上で相互に連携し依存する情報コンテンツの集合において、コンピュータによる解釈実行が可能な形式で記述された要素間の依存関係を伴うウェブ・ドキュメントの集合であるコンテンツ・ネットを備え、一部の要素にコンテンツ変更が生じた場合、変更された要素に依存するすべての要素をコンピュータによって自動的に連鎖更新する。

【選択図】　図１２

認定・付加情報

特許出願の番号	特願２０００－０９１４５５
受付番号	５００００３９００９６
書類名	特許願
担当官	第三担当上席　　００９２
作成日	平成１２年　４月　３日

＜認定情報・付加情報＞

【提出日】　平成12年 3月29日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔５００１４００８０〕

１．変更年月日　　２０００年　３月２９日
［変更理由］　新規登録
住　所　　新潟県南魚沼郡大和町大字市野江甲３５
氏　名　　若山　俊弘